

Date, Time, Name, Type, Status

"4/19/2003", "11:20:35", "Robert", "Web Accept Payment Received", "Uncleared"

"4/17/2003", "14:30:26", "Tanya", "Web Accept Payment Received", "Completed"

"4/16/2003", "6:38:28", "Mark", "Web Accept Payment Received", "Completed"

"4/14/2003", "19:54:34", "Tigor", "Web Accept Payment Received", "Completed"

"4/12/2003", "16:41:08", "artem", "Web Accept Payment Received", "Completed"

+ record

- record

- 4/19/2003

- 11:20:35

- Robert

- Web Accept Payment Received

- Uncleared

+ record

- record

- 4/16/2003

- 6:38:28

- Mark

- Web Accept Payment Received

- Completed

+ record

- 4/14/2003

- 19:54:34

- Igor

- Web Accept Payment Received

+ record "Robert"

+ record "Tanya"

+ record "Mark"

+ record "Igor"

- 4/14/2003

- 19:54:34

- Igor

- Web Accept Payment Received

- Completed

```
using System;
namespace IVDEV.Trees
{
    /// <summary>
    /// Summary description for MyControl
    /// </summary>
    public class MyControl : System.Windows.Forms.UserControl
    {
        private void MyTabPage_Enter(object sender, EventArgs e)
        {
            rb = new RepresentationBuilder();
            rb.code = textBox1.Text;
            rb.path_to_grammar = "C:\\grammars";
            rb.BuildGrammarTree();
            if ( a==1 )
            {
                b=2;
                while ( a==1 ){ a+=2;}
            }
            //the rest of the code is omitted for clarity...
        }
    }
}
```

```

using System;
namespace IVDEV.Trees
{
    /// <summary>
    /// Summary description for MyControl
    /// </summary>
    public class MyControl : System.Windows.Forms.UserControl
    {
        private void MyTabPage_Enter(object sender, EventArgs e)
        {
            rb = new RepresentationBuilder();
            rb.code = textBox1.Text;
            rb.path_to_grammar = "C:\\grammars";
            rb.BuildGrammarTree();
            if ( a==1 )
            {
                b=2;
                while ( a==1 )
                {
                    a+=2;
                }
            }
            //the rest of the code is omitted for clarity...
        }
    }
}

```

```
<?php
```

```
echo "Example statement\n";
```

```
if($ a == 1)
```

```
{
```

```
echo "value in a is 1\n";
```

```
}
```

```
function foo ($ arg_1, $ arg_2, ..., $ arg_n)
```

```
{
```

```
    echo "Example function.\n";
```

```
    return $retval;
```

```
}
```

```
?>
```

```
<?php
echo "Example statement\n";
if ( $a == 1 )
{
    echo "value in a is 1\n";
}
function foo ( $arg_1 , $arg_2 , ... , $arg_n )
{
    echo "Example function.\n";
    return $retval;
}
>>
```

```
if (boolean_expression)
```

```
    embedded_statement
```

```
else
```

```
    embedded_statement
```



```
if (a==1)
{
    b=2;
}
else
{
    c=3;
}
```

||| boolean_expression |||

{
 embedded_statement
}

||| else |||

{
 embedded_statement
}

→ if (a==1)

 b=2;

→ else

 c=3;

if (a==1)

... b=2;

0. File

0. Line

0. Value

0. Line

0. Value

0. Line

0. Value

rule representation mapping

⊖ "if" "(" boolean_expression ")"
└── embedded_statement
⊖ "else"
└── embedded_statement

representation of an if_statement

⊖ if (a==1)
└── b=2 ;
⊖ else
└── c=3 ;

"if" "(" boolean_expression ")"
└── embedded_statement
⊖ "else"
└── embedded_statement

⊖ if (a==1)
└── b=2 ;
⊖ else
└── c=3 ;